1. **Understand the Problem**:
   * Carefully read and comprehend the problem statement, ensuring you fully grasp its requirements.
2. **Examine Test Cases**:
   * Review provided test cases to understand the expected input and output formats.
3. **Validate Input and Output**:
   * Confirm data types, ranges, and formatting requirements for both input and output.
4. **Decompose the Problem**:
   * Break down the problem into smaller, manageable sub-problems to simplify the solution.
5. **Select Data Structures**:
   * Choose appropriate data structures (arrays, linked lists, trees, etc.) based on problem requirements.
6. **Design an Algorithm**:
   * Develop a step-by-step algorithm to address the problem, considering various approaches.
7. **Create Pseudocode**:
   * Write pseudocode to outline your solution and structure your thoughts before coding. Write steps (as comments) that can have code below them.
8. **Implement Your Code**:
   * Begin coding based on your pseudocode, testing and debugging along the way to catch errors.
9. **Optimize Your Solution**:
   * Look for ways to optimize your code, such as reducing time or space complexity.
10. **Thorough Testing**:
    * Rigorously test your code with different inputs, including edge cases and boundary conditions.
11. **Submit and Validate**:
    * Submit your solution to LeetCode once confident it's correct, ensuring it passes provided test cases.
12. **Learn from Mistakes**:
    * If your solution doesn't pass all test cases, analyze your code, understand the mistakes, and learn from them. And, refer to the [LeetCode Guide](LeetCode%20Guide.DOCX) for guidance.

Top of Form

Bottom of Form